By Patrick R. Burasa
Published: 2008-05-12 20:30

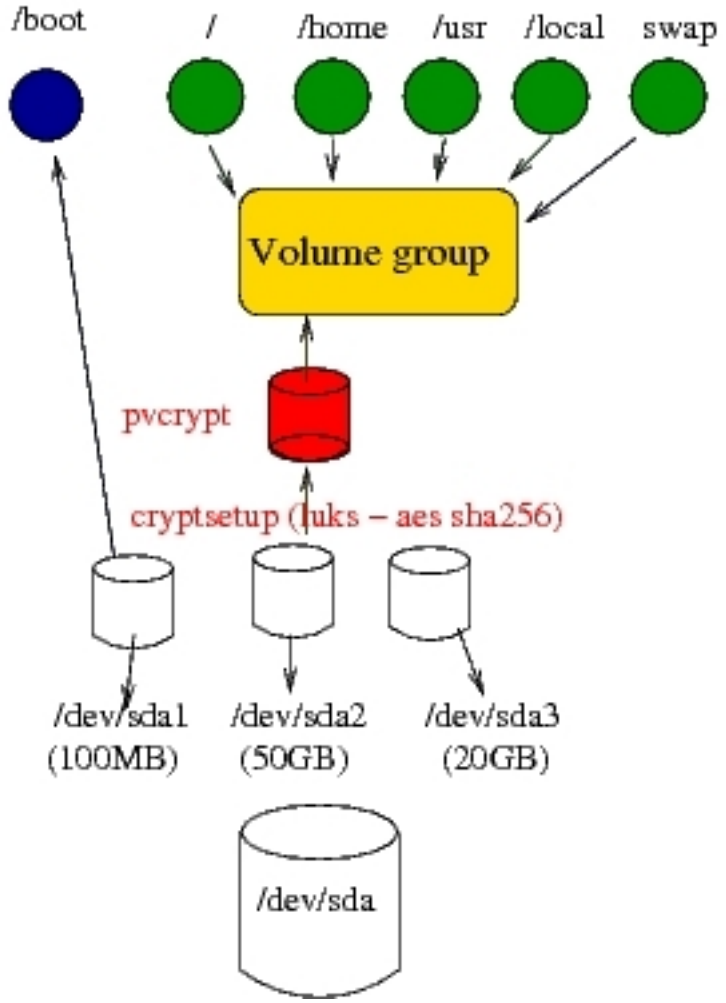
Encrypted Root LVM

I am assuming that you already know how to set up an encrypted file system using cryptsetup with luks (or something else). There are several howtos. I am also assuming that you are familiar with LVM2.

This tutorial deals only with how to add an extra encrypted physical volume to a volume group pool containing other encrypted physical volumes. This is typical scenario if, at first, you have set up your encryption at a physical partition level (/dev/sdaX where X is the a number of your partition), then you setup your LVM on top of the encrypted partition. If at some later time you want to add another partition in your volume group, you will also want to have it encrypted in order to maintain the same level of security. In order for your machine to boot, initramfs needs to be able to unlock both PVs in order to reconstruct the entire volume group where your root lv is lying.

Some few weeks ago I installed Ubuntu (Hardy Heron 8.04) on mythinkpad T60. My boss recommended that I set it up with encrypted LVM(at least for the home logical volume). I used a tutorial at **https://help.ubuntu.com/community/EncryptedFilesystemLVMHowto**.

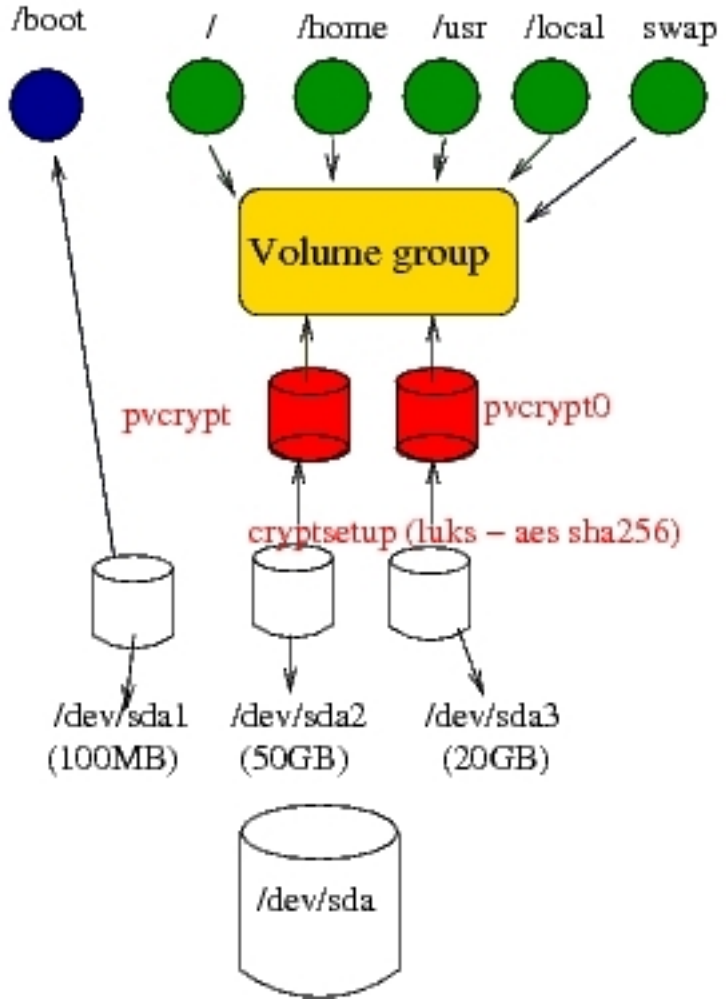
After I was done, I ended up with something looking like this:



As you can see in the figure above, everything is encrypted except the /boot partition. This setup worked fine. When I booted after initramfs loaded my kernel image, it proceeded to prompt for the password to unlock pvcrypt. Without this step, it wouldn't be able to decrypt and mount the root logical volume.

A problem arose as soon as I thought it was time to add the remaining 20GB on /dev/sda3 into my volume group. The issue is pvcrypt (/dev/sda2) is

encrypted, and it will break my whole security if I just created a physical volume from /dev/sda3 and then add it to the volume group. Why? Because it wouldn't be encrypted and the data that would end up on it from the logical volumes wouldn't be encrypted. The quickest way to deal with the issue was to encrypt it too as pvcrypt0 then I would add it to my volume group and end up with something like in the following figure.



The figure above shows my lvm setup after adding the 20GB partition in the volume group as an encrypted pv (pvcrypt0).

The problem with this is that initramfs needs to be told to ask for the second password in order to unlock pvcrypt0 and reconstruct the volume group. If you don't do this, initramfs will only prompt for the password of pvcrypt but it will ultimately fail to boot since it won't be able to unlock the pvcrypt0.

First examine the file *cryptroot* in your kernel image. To do this unpack your kernel image in a directory:

```
mkdir /tmp/foo

cd /tmp/foo

zcat /boot/initrd.img-$(uname -r) | cpio -iv

cat conf/conf.d/cryptroot
```

Your cryproot file should look like this:

```
target=pvcrypt,source=/dev/sda2,key=none,lvm=vg0-root,lvm=vg0
```

Obviously, you won't be prompted for pvcrypt0 password. To tell initramfs to ask for the second password at boot time, you need to add it to the cryptroot (eventually create it) file in */etc/initramfs-tools/conf.d:*

```
cd /etc/initramfs-tools/conf.d

vim cryptroot
```

Add this this line and save the file.

```
target=pvcrypt0,source=/dev/sda3,key=none,lvm=vg0-root,lvm=vg0
```

Now run update-initramfs (update the kernel boot image(s)):

```
update-initramfs -k all -u
```

When this is done, you can now check if your updated kernel image knows about pvcrypt0:

```
cd /tmp/foo

zcat /boot/initrd.img-$(uname -r) | cpio -iv

cat conf/conf.d/cryptroot
```

This time you should get two lines:

```
target=pvcrypt,source=/dev/sda2,key=none,lvm=vg0-root,lvm=vg0
target=pvcrypt0,source=/dev/sda3,key=none,lvm=vg0-root,lvm=vg0
```

Et voilÃ . When you reboot. Initramfs will ask for two password. It will be nice if there was a way of telling initramfs to attempt to use the same password. A simple way to not have to type the second password would be to actually have it in a key file.